

Student Name:

Stud id:

sect #: serial#:

University of Bahrain

College of Information Technology

Department of Computer Science

ITCS332: Concepts of Programming Languages

Quiz#4: Chapter 5_Names

QUESTION ONE: Fill in blanks

[4 pts]

- 1) In C++, the “||” symbol is bound to “or” operation during the **language design** time. The internal representation of numbers is bound to 2’s complement during the **language implementation** time.
- 2) In languages that use dynamic type binding, the type of a variable is specified using a **referencing context** or whenever **it is assigned values during execution**.
- 3) In C++, the **lifetime** of a variable defined inside a block begins whenever **the block is entered** and ends whenever **the block is exited**.
- 4) The lifetime of an explicit-heap dynamic variable begins whenever **the defining statement/operator is executed** and ends whenever **the deallocating/deleting statement is executed**.

QUESTION TWO: carefully study the following C++ code and answer the questions below:

[7 pts]

```
50) int x = 6, y = 4, f = 2*x ;
51) void main ()
52) { static double t = -7.5;
53)     int x = 7, y = 11, f = 3*x;
54)     { //block #1
55)         int f;
56)         cout << y << '\t' << f << '\t' << t << endl;
57)         { //block #2
58)             int y = 30;
59)             int x = 20;
60)             cout << x * y;
61)             .....
62)         }
63)         x--; y *= 2; t *= 4; f = x - y;
64)         .....
65)     }
66) }
```

- 5) The scope of the variable x defined in line#53 is **from line #53 to line #66 except lines #59 to #62**.
- 6) The referencing environment at line#64 is **f of block #1 and x, y, t of main**.
- 7) The referencing environment at line#61 is **x,y of block#2, f of block#1, and t of main**.
- 8) The lifetime of the variable t defined in line#52 begins when **The function main is loaded** and ends when **The function main is exited**.

Student Name:

Stud id:

sect #: serial#:

University of Bahrain

College of Information Technology

Department of Computer Science

ITCS332: Concepts of Programming Languages

Quiz#4: Chapter 5_Names

QUESTION ONE: carefully study the following C++ code and answer the questions below: [7 pts]

```
20) int x = 4, y = 6, f = 3*x ;
21) void main ()
22) { static double t = -4.5;
23)     int x = 11, y = 7, f = 2*x;
24)     { //block #1
25)         f++;
26)         cout << f << '\t' << t << endl;
27)     } //block #2
28)         int y = 20;
29)         int x = 30;
30)         cout << x * y;
31)         cin >> x;
32)         .....
33)     }
34)         x++; y *= 2; t *= 2; f = x + y;
35)         cout f << '\t' << t << endl;
36)     }
37)         .....
38) }
```

- 1) The scope of the variable x defined in line#23 is **from line #23 to line #38 except lines from #29 to #33.**
- 2) The referencing environment at line#32 is **x,y of block#2 and t,f of main.**
- 3) The referencing environment at line#37 is **x,y,t,f of main.**
- 4) The lifetime of the variable y defined in line#28 begins whenever **The block #2 is entered** and ends when **The block #2 is exited.**

QUESTION TWO: Fill in blanks

[4 pts]

- 5) In static type binding languages (such as FORTRAN), the type of a variable is specified using **explicit declaration statement** or **implicit declaration.**
- 6) The main disadvantage of stack-dynamic variables is **Time overhead for allocation and deallocation.** The main advantage of stack-dynamic variables is **Efficient use of memory space.**
- 7) In programming languages, a data type specifies: **range of possible values assigned to variables** and **a set of operations applicable on those variables.**
- 8) Static variables are allocated storage when the **defining program unit/block is loaded for the first time** and deallocated when **the entire program unit/block is exited.**

- 9) In a static-scoped language, the initializing expression of a variable must contain **constants/predefined constant names**. In a dynamic-scoped language, the initializing expression of a variable must contain **variables/constants**.
- 10) **Static variables** are bound to storage during the **LOADING** time. A data type such as **bool** is bound to a range of possible values during the **LAMGUAGE IMPLEMENTATION** time.
- 11) **For every assignment to a subrange** variable, the **RANGE CHECKING** is done at run-time, and the **TYPE CHECKING** is done at compile time.
- 12) In a static-scoped language; referencing environment depends on **the textual layout** of program unit/blocks. In a dynamic-scoped language, the referencing environment depends on **the calling sequence** of program units.

Write a C++ code that illustrates the use of stack-dynamic variables.

Write a C++ code that illustrates the use of explicit heap dynamic variables.